Data processing device, data structure and method for delivering financial information for a financial information database

The present invention comprises a data processing device programmed for creating and distributing financial information in accordance with the features of the precharacterizing part of claim 1 and also to a hierarchical data structure for messages for the financial sector and a method for creating and delivering financial information for the financial sector in accordance with the features of the precharacterizing parts of claims 8 and 13.

Known data processing devices, data structures and methods for delivering financial information are limited to piecemeal formalization of the financial information. Many data structures are designed primarily for one particular processing purpose. Thus, known data structures and methods are normally oriented to covering individual ones of the various areas such as stock exchange, securities trading, management of securities and loans, securities assessment, securities handling, securities holding or tax declarations. In addition, many data processing devices encounter the difficulty that the connection cannot be set up systemically by individual data elements from the data structure.

The known data processing devices, data structures and methods thus allow neither full nor automated processing and storage of financial data in a database while maintaining referential integrity.

The following example clarifies the situation. A customer, e.g. a manager of securities and loans, manages a particular share, for example. For this share, the data supplier provides him with basic data, exchange rate data, data about dividend payment and other data connected directly to the share. In the event of a capital increase which is distributed in the form of a right to a new share, it is now necessary for the basic data of the right and of the new share to be delivered as well, however. Otherwise, automatic processing in the environment of deposits management or management of securities and loans is not possible. In the case of the known data processing devices and data supply systems, the customer today normally needs to obtain the supply of the data for the right and for the new share himself, manually from the data supplier. This manual activity complicates or prevents reaction-speed management of securities and loans and the efficient creation of portfolio information, such as deposits statements, securities and loans statements for tax declaration etc.

Fragmentation into a large number of data structures in line with the aforementioned areas, as is done by the known data processing devices and methods, does not cover a semantically appropriate and technically correct order for the data elements which are supplied as messages being maintained. However, systematic and efficient processing requires the data to be able to be recognized and processed directly as new data, change data or deletion data. The known data processing devices provide no data structure from which this can be recognized for each individual data field. As described above, known data processing devices therefore require basic data for new instruments obtained as a result of a distribution, for example, to be ordered manually from a data supplier in a separate work step, for example. This means that the known data processing devices do not allow any automatic guarantee of referential integrity without manual intervention by the customer or the supplier.

Most consumers of financial information are interested in a subset of information, for reasons of faster processing. A subset is usually defined as a list of financial instruments, as a list of companies, as a list of market places and markets or as a type of financial instrument or as combinations thereof. There is no known data processing device, data structure or method which can take a subset definition of this kind as a basis for automatically creating a completion which results in a minimal but complete set of financial information data.

It is the technical object of the present invention to provide a data processing device, a data structure and a method for computer-aided creation and delivery of messages containing financial information about financial instruments, institutions, stock exchanges, company events, payments etc., so that fast, efficient and, as far as possible, completely automatic processing and storage of financial data is possible in a database while maintaining data consistency.

This object is achieved by a data processing device in accordance with patent claim 1, by a data structure in accordance with patent claim 8 and by a method in accordance with patent claim 13.

The inventive data processing device for delivering financial information for a financial information database allows commercially available computer and programming systems to be used to process the various types of financial information in a novelly simple fashion. In this context, the inventive data structure supports systematic, automatic processing by virtue of its normalization for all data elements of financial information to a generic data type. Customers, consumers of financial information, normally differ significantly through their various needs according to data width and data depth. The inventive method conditioning the data provides a simple way of creating individual data sets. The method guarantees that the completeness and correctness of the data are ensured. Completeness means that no data are missing which are required for automatic processing and/or whose absence would result in a gap in information. The aforementioned completeness is also referred to as semantically complete or as semantically appropriate. A semantically appropriate order of messages exists when each subset of messages which has been delivered at a particular time represents a complete data set.

The invention provides a data structure for the messages which puts business-related aspects of financial information, such as basic data from institutions, basic data from financial instruments, payments from financial instruments etc., into one or more respective data substructures, known as data elements. The data structure is designed such that it represents complete formalization of all available and conceivable information. That is to say that the available data units and data fields of the data elements cannot be broken down or formalized further in appropriate fashion. The data elements contain what are known as key values, which comprise one or more data fields. These key values allow distinct identification of specialist and technical entities. The data elements are linked to one another by key values, which are part of the data elements, such that each data element has a particular set of dependent data elements, depending on stipulated method logic. This means that all real-world, that is to say semantically appropriate, connections between two data elements can be recorded. By way of example, a key value appearing in the data element in which an option is depicted is used to record the link between the option's data element and the issuer's data element for a particular issuer, the issuer's data element having recorded the business form as issuer. That is to say that there is a continuous concept of specialist and nonspecialist keys which allows all real-world connections between the data elements.

The inventive method allows initial, "inventory" data elements, on the one hand, and mutation-oriented "delta" data elements, on the other hand, to be produced. An associated data processing device which has been programmed such that the inventive method can run on it produces from a source data inventory one or more files and/or a data stream of inventory data elements and/or delta data elements which are transmitted to the customer.

The data elements arising in the files produced or in the data stream are conditioned such that they can be identified as inventory data elements or delta elements, are completely provided with key values and are organized. The organization of the data elements which is produced by the method guarantees that dependent data elements always appear after the data elements on which they are dependent. This ensures that, following delivery of the inventory and/or delta data elements in the form of files and/or in the form of a data stream to customers, linear, automatic processing on a computer never encounters key values which point to as yet unprocessed data elements.

The particular feature of the delta data elements is the flag "deletion", "mutation" or "new". In the case of deletion and mutation, both the updated data and the old, no longer current, data are delivered completely too, so that the changes in content can be recognized without being assisted by access to a database.

The inventive data structure is distinguished further by virtue of it being able to contain what are known as informative data, which are not taken into account by the automatic, computer-based processing but improve legibility for humans. By way of example, in the case of data elements for a share which are associated with the financial instruments area, the key value of the company for which the company is the issuer of the share is included. So that the data element for the share is more readable, the company name is also supplied as an informative segment. The company name is delivered for automatic, computer-based processing with a data element from the institutions area.

The method allows not only the formation of inventory and delta data elements from an available source data inventory but also the creation of inventory and delta data elements on the basis of a subset definition which contains a list of key values. By way of example, the subset definition contains a list of instrument identifications (key values for financial instruments), a list of markets or a list of financial instrument types or combinations thereof. In addition, the lists may specify parameters for the key values which are made possible by a further specification of the data associated with the subset. The data processing device can use method-based programming at any time to create, for every such subset definition, data element subsets containing inventory and/or delta data elements which have the aforementioned property of completeness. The data element subset created is minimal, however, which means that it is not possible to remove a data element without losing fundamental information relating to an element from the subset definition or without the completeness property still applying. At minimum, the property allows the processing involvement for the data to be kept minimal. The organization of the data elements in the files and/or in the data stream also allows the linear, direct processing for this data element subset, as illustrated above, without manual intervention while maintaining referential integrity.

The invention comprises a data processing device which is programmed to create and distribute financial information and which has at least one network-compatible interface. The data processing device can be connected to a dynamic number of customer systems via the at least one interface such that the at least one interface can be used to transmit financial information to the customer systems from the data processing device at any time. In addition, the at least one interface is in a form such that, conversely, the customer systems can transmit lists containing key values and parameters, such as market place and the like, to the data processing device, the supply files and/or supply data streams created comprising inventory and/or delta data elements.

To be able to ensure the security of the data better, it is advantageous to design the at least one interface as a protected interface. If the at least one interface can be used both for actively fetching the data from the customer systems, known as a pull service, and/or sending them, known as a push service, without active participation by the customer systems, e.g. through time-controlled delivery of the data, then a high level of flexibility for the data supply and better dealing with the needs of the customers are possible.

In one particularly preferred embodiment, the data processing device is designed such that three data processing systems are recognizable which are preferably equipped with a respective processor and a respective data store. These data processing systems are connected to one another by means of network and are programmed such that a data collection system with its own database, an inventory data generation system with an inventory database and, depending on the embodiment, possibly also with a delta database, and also a customer distribution system with at least one interface and at the outside (see above) a delta database are recognizable. A data processing device of this kind is preferably developed such that the customer distribution system can create semantically complete supply files and/or supply data streams containing inventory and/or delta data elements for an unlimited number of lists containing primary keys and parameters from the institutions and financial instruments areas. These inventory and/or delta data elements in the supply files and/or supply data streams have been sorted such that the customer systems which can be connected to the data processing device can, provided that they are equipped with their own data store and their own data processing, process the data very quickly without additional database access operations, and the referential integrity of the data stores of the customer systems is maintained at all times.

It is also advantageous if the inventory data generation system in such a data processing device creates inventory data elements upon request or under time control from the database provided by the data collection system and stores them in the inventory database provided for this purpose. The associated delta data elements are either created immediately when the inventory data elements are stored, and are stored in a further "delta database" and the data elements which the customer requires are then delivered to the customer. Another possibility is for the delta data elements to be generated and sent only upon delivery to the customer. These delta data elements can then be stored in a delta database, which is not absolutely necessary, however. If the delta data are not stored, there is advantageously no storage location required either. In this case, the data conditioning takes place dynamically. Regardless of whether or not the delta data are stored, the inventory data elements and/or delta data elements conditioned in this manner are transmitted to the customer via the at least one interface, preferably at particular times.

In addition, one or else more interfaces can also be provided. Thus, a second interface may be appropriate, for example, which permits direct access by the customer systems to particular data in the data collection system in order to allow the customers to access data recently captured in the data collection system in real time when the conditioned data are transmitted periodically.

It is possible to position the individual data processing systems which are combined in the data processing installation described above in greater or lesser local proximity. The data processing systems can then sometimes share processors and/or data stores in any combination. Alternatively, it is possible to position these data processing systems so that they are physically separate and are at a greater or lesser distance from one another. It is then appropriate for each data processing system to have its own processor and its own data store. It is likewise conceivable for two data processing systems to be positioned physically close to one another but separate from the third data processing system, with the relevant design option regarding data store and processor.

Particularly advantageously, the data processing device is operated with a method in which inventory and delta data elements are delivered in a particular message order. In this context, the message order supports data management with referential integrity such that every data element is processed on the basis of its position in the message order, and in this way the referential integrity of the data management remains assured.

The method is particularly advantageous when it is applied to a data structure, as described further below. It is then possible for predetermined method steps to form inventory and/or delta messages from any desired subset of primary keys from the institutions and financial instruments areas, the property of the referential integrity remaining assured and the volume of messages remaining minimal.

It is likewise very advantageous if, based on the interactive selection of parameters, key lists are extended automatically, since in this way it is possible to create portfolio information automatically.

It is very advantageous if the method is applied to an inventive hierarchical data structure for messages for the financial sector. The inventive data structure has the following message flags: inventory and delta. In addition, the message order is based on the five areas comprising the metadata, institutions, financial instruments, events and prices areas. The areas or keys of the data elements in this data structure can be described as follows:

the foreign keys available in the institutions area, which are unable to be resolved locally, can be resolved in the metadata area. The foreign keys available in the financial instruments area, which are unable to be resolved locally, can be resolved by the metadata and institutions areas. The foreign keys available in the events area, which are unable to be resolved locally, can be resolved by the metadata, institutions and financial instruments areas. The foreign keys available in the prices area, which are unable to be resolved locally, can be resolved by the metadata, institutions, financial instruments and events areas.

A computer program having a program code which, when executed on a data processing installation, results in the above-described method for creating and delivering financial information being carried out, can advantageously be used in the financial sector.

A computer program of this kind or a program code of this kind can be stored on any desired data storage medium.

Further advantageous embodiments of the data processing device, its data processing systems, the method and the data structure etc. can be found in the patent claims.

The inventive data processing device, the inventive structures and the inventive method are explained by way of example with reference to the figures below, in which, purely schematically:

figures 1a and 1c	show an overview of the data processing device for creating and delivering inventory and delta data elements;

figure 1b shows a second embodiment of the data processing device for creating and delivering inventory and delta data elements;

figure 2 shows the inventive data structure for messages containing financial information;

figure 3 shows four structural areas of the messages;

figure 4 shows the illustration of an indirect relationship between two data elements;

figure 5 shows a schematic illustration of the completeness formation with primary keys from an area B2;

figure 6 shows a schematic illustration of the completeness formation with primary keys from an area B3;

figure 7 shows a schemati c illustration of the combination of the completeness formations with primary keys from the areas B2 and B3, and

figure 8 shows a schematic illustration of the automatic key extension through interactive choice of data element types.

Figures 1a and 1c show an overview of an inventive data processing device 10 for creating and delivering financial data by way of example. In particular, they show the creation of inventory and delta data elements and the distribution of financial information over customer systems. In addition, figure 1b shows a variant of the inventive data processing device 10 for creating and delivering financial data. A common feature of the two variants is that in a first step electronic data sources (not described in more detail)

- two sources SYS1 and SYS2 serve as an example in this case - deliver financial data in different, nonstandard formats via a commercially available data network NET to a data collection system HPS in the data processing device 10. The data processing device 10 with the data collection system HPS is equipped with a database HPSDB and is programmed such that a complete inventory data element set can be created at any time. The data collection system HPS may be a central data processing station which formats the incoming data, inter alia, for the database HPSDB, or it may comprise formatting stations in situ which format incoming data and deliver them to the database HPSDB for processing and forwarding. In this case, the set of data elements is preferably classified into 5 areas B1 to B5 of an inventive data structure, as is shown further below by means of figure 2. The use of a LogFile allows the data changed during a particular period to be filtered out of the total data set stored in the database HPSDB at any time.

An inventory data generation system as part of the HPS creates complete, new inventory data elements provided with data from the data elements stored in the database HPSDB upon request or by means of time control and preferably taking into account the LogFile. The new inventory data elements are stored in the inventory database SupplyDB. A renewal process is used to check whether an inventory data element already appears in the inventory database SupplyDB. If this is the case then each segment of the new data element is compared with the segments of the previous one and a delta data element is produced if differences are found. A delta data element contains all new, altered, deleted and informative data segments and also all unaltered data segments and the data segments which are no longer current, flagged as old. The checks with their associated creation of delta data elements can take place, as one alternative, when the inventory data elements are created, in which case the delta data elements are then immediately stored in a delta database DeltaDB, cf. figure 1b. Alternatively, it is possible for the check not to take place until the inventory data elements are delivered, in which case the delta data elements are preferably likewise stored in a delta database DeltaDB for gapless documentation, cf. figures 1a and 1c. In a further process, a check is periodically performed to determine whether data elements are also present in the database HPSDB for all the data elements which are present in the inventory database SupplyDB. Should a data element in the inventory database SupplyDB not be present in the database HPSDB then a deletion delta data element is produced. The data element which is still present in the inventory database SupplyDB is deleted. The inventory data elements are respectively delivered with the associated delta data elements via a customer distribution system VDFS.

The customer distribution system VDFS can be connected to a commercially available data network NET via an interface S. In the example shown in figure 1c, the customer distribution system is connected to three customer systems CS1 to CS3 via its interface S and via the data network NET. However, any number of customer systems CS1 to CSn may naturally be connected to the customer distribution system VDFS, and there may also be more than one interface. The customer systems CS1 to CS3 transmit lists Cifps1, Cifps2, Cifps3 containing the desired key values and parameters to the data processing device 10. For each customer-specific key list Cifps1 to Cifps3, the customer distribution system VDFS creates primary keys and supply files or supply data streams CL1 to CL3 containing inventory and/or delta data elements from the total set of data elements generated by the delta data generation system VSS. As described further below, this is done using six method steps for institutions Si1-Si6, for example, using five method steps for financial instruments Sf1-Sf5, for example, and using a combination method step Sc1, for example. The supply data or supply data streams CL1 to CL3 created on the basis of the customer-specific lists Cifps1 to Cifps3 are delivered to the customer systems CS1 to CS3 via the interface S and the data network NET. The customer systems CS1 to CS3 are now able to supply inventory or delta data elements to their database systems CDB1 to CDB3 directly, automatically and without the need for manual action, while maintaining referential integrity. The data state in the customer databases is therefore such that all dependent data elements have all the data elements available on which they depend. By way of example, this ensures that in the event of a company merger the data about all the companies involved, about the financial instruments which are relevant to the event and about the payments which are relevant to the event are selected and delivered by the data processing device with the customer distribution system VDFS and that the processing can take place on the customer systems CS1 to CS3 automatically and without manual action.

The data processing device can be classified schematically into three data processing systems A1 to A3: the first data processing system A1 with the data collection system HPS, the dedicated database HPSDB; the second data processing system A2, which in a first variant has the delta data generation system VSS with the inventory database SupplyDB and the delta database DeltaDB, cf. figure 1b; and in a second variant comprises just the delta data generation system VSS with the inventory database SupplyDB and, cf.

figures 1a and 1c; and the third data processing system A3, which comprises a customer distribution system VDFS with at least one interface S and, depending on whether or not the second data processing system comprises the delta database DeltaDB, also a delta database DeltaDB. If the delta data elements are not produced until the inventory data elements are delivered then there does not actually have to be any delta database DeltaDB at all. The data processing systems A1 to A3 preferably each have a processor and a data store themselves and are connected to one another by means of network, as shown in figure 1a. Depending on whether the data processing systems A1, A2, A3 are provided locally at one location or at locations which are a long way away from one another, the network may be an internal or an external network. If the data processing systems A1, A2, A3 have been provided at locations which are together then they can also share data stores and/or processor(s).

As mentioned above, the set of data elements is preferably classified into five areas B1 to B5 of an inventive data structure. Figure 2 now shows the five areas within the inventive data structure. The metadata area B1 contains structure and reference data for all financial information entities. The metadata determine the tree structure for all data element types and identify which keys are primary or foreign. The reference data contain value ranges which are relevant to all data elements, such as country codes and language codes. The institutions area B2 comprises financial information about companies of a wide variety of types. This financial information contains, inter alia, basic company data relating to joint stock companies, basic data relating to countries, international organizations, data about company relationships, such as holding structures, and data about rating information. The financial instruments area B3 comprises financial information about financial instruments. This information includes, inter alia, basic data about shares, obligations, interest, options, futures, government bonds and stock exchange indices, data about financial instrument company relationships, data about issues with their conditions, data about financial instrument structures and dependencies, data about coupon definitions, data about the capital structure of the financial instruments, data about guarantee and supply regulations, data about ratings and oppositions. The events area B4 comprises financial information about finance-related events. These events include, inter alia, basic data about company events, payment events and trade events. The data about company events include data about capital changes, company mergers and takeovers, transformations of financial instruments and legal events. The data about payment events comprise data about payments, repayments, amortizations, taxes in connection with financial instruments and with national and international tax legislation, distributions and subscription rights. The data about trade events comprise data about listings, delisting and trade-related information. The prices area B5 comprises valuation rates and tax rates (relevant rates for tax declarations), market rates, valuations, prices, etc. The lists for the individual areas B1 to B5 are not conclusive and can be expanded to suit the current situation. Any time a new financial instrument is introduced it is thus possible to incorporate it into the financial instruments area B3.

The five areas and their dependencies among one another are shown in figure 2. An area BY is dependent on an area BX precisely when there is at least one data element in the area BY which contains one or more foreign keys whose associated primary key appears in one or more data elements from area BX. The result of the area formation defined in this manner is now a noncircular dependency (A1-A10) for the five areas B1 to B5, in line with the invention. That is to say that none of the areas is dependent on itself over other areas.

Figure 3 shows two data elements DE1, DE2 from t he inventive data structure by way of example. Each data element DE1 to DEn has a data element name. By way of example, basic data for institutions are held together in a data element "TKIAD" and basic information for a financial instrument is held together in a data element called "TKFAD".

Every data element also contains a tree-like hierarchy of different data segments SEG_H, SEG_I and SEG1 to SEGn. The first data segment SEG_H, called head segment, contains all-embracing information for the whole data element. The second data segment SEG_I, called key segment, contains a distinct identification for the data element. From data segments SEG1 to SEGn, each superior segment contains information, particularly key values, for the segments which are directly or indirectly dependent on it.

The first data element DE1, shown in figure 3, has a head segment SEG_H, a key segment SEG_I and, by way of example, further data segments SEG1 to SEG7. In this example, for the sake of simplicity, the data element DE2 has just one head segment SEG_H, a key segment SEG_I and two further segments SEG8, SEG9. The head segment SEG_H contains the data fields data element type, creation date, generation number and a flag for inventory or delta data element SD-FLAG (not shown explicitly). The data element type identifies the type to which the data element belongs. The metadata, likewise a set of data elements, determine for each type of data elements the tree structure, the position and the number of data segments and also the design of the data segments. The generation number is a version identifier which is used to identify data elements from various production runs. The key segment SEG_I comprises an identification number which distinctly identifies the data element together with the generation number and the creation date.

Every further data segment after the head segment SEG_H and the key segment SEG_I contains a segment name SN, a processing instruction VA and a list of data fields. The segment name identifies the segment type distinctly. The metadata are therefore used to define the number and type of fields. The data fields can be split into single and combined data fields. Single data fields SF contain a single value, such as company name. Combined data fields CF comprise at least two values, which may be a currency and a numerical sum value in the case of a cash payment, for example.

The processing instruction VA is a list of flags which can contain a "key", "inventory", "deletion", "mutation", "new", "old", "unaltered" or "informative" flag for each value in a data field. The "key" flag means that the field value belongs to a primary or foreign key. The "deletion" flag means that the field value no longer exists. The "mutation" flag means that the field value has been changed. The "new" flag means that the field value is a new, as yet undelivered value. The "old" flag means that the field value is no longer current. The "unaltered" flag means that the field value is unaltered. The "informative" flag means that the field value is redundant and is not intended for electronic processing. Inventory data elements have exclusively the flags "key", "inventory", "novel" and "informative" in their processing instruction. Delta data elements, on the other hand, use all the aforementioned flags in the processing instruction.

In the example shown in figure 3, SB is a key dependency from the data element DE2 to the data element DE1. The dependency exists because the segment SEG3 with its dependent segments SEG5, SEG6 and SEG7 contains key values and data from an information entity such as a joint stock company. The data element DE2 uses the key values contained in the SEG8 to set up a link to this information entity. The key values from the information entity are called primary keys in SEG3. Primary and foreign keys can be distinguished using the metadata. If the key values in a dependent data element are used to link information entities, the key values are called foreign keys. A foreign key in a data element can be resolved if the primary key appears further up in the segment hierarchy of the data element itself or in another, available data element as a primary key. The referential integrity is maintained for electronic processing of a list of data elements precisely when each foreign key appearing in the data elements can be resolved by its own data element or by at least one data element preceding in the list or by a data element from a database provided for this purpose.

If a foreign key is resolved within the data element in which it appears or by a data element from the area of the data element carrying a foreign key then the foreign key is called locally resolvable. By way of example, a data element from the events area B4 which represents the payment of a dividend contains a segment with a foreign key from a share. The share has an instrument basic element "TKFAD" from the financial instruments area which contains the appropriate primary key in one of its segments. The data element with the dividend payment is therefore dependent on the data element "TKFAD".

Figure 4 illustrates the relationship directly and indirectly dependent between two data elements. A data element DEy is directly dependent (dir) on a data element DEx if DEy has a foreign key which appears in DEx as primary key. DEz is indirectly dependent on DEx if DEz is directly dependent on DEy and DEy is directly dependent on DEx.

Another part of the inventive method is completeness formation. Completeness formation denotes a method which takes a prescribed specification for a subset and forms an expansion set, so that no data are missing which are required for automatic processing and/or whose absence would give rise to a gap in the information. By way of example, a dividend payment in the form of a right also always includes the basic data about the right. Otherwise, the data for the dividend payment are incomplete and cannot be processed correctly and hence are of no benefit for business use. Completeness formation is intended particularly for delivery to customers who have different needs according to data depth.

Figure 5 schematically shows the completeness formation for inventory data elements in the case of a prescribed list of primary keys Ips (institution primary key) from the institutions area B2. The inventory areas Supply1 to Supply4 are those inventory data sets which contain the data elements belonging to the areas B1 to B4. In a first step Si1, the inventory data elements from the inventory area Supply2 of the institutions area B2 which contain one or more of the keys from the list Ips as primary key are selected. In the same step Si1, these inventory data elements have directly and indirectly dependent inventory data elements from the inventory area Supply2 added to them, so that the subset T2Si1 is produced. In a further step Si2, the inventory data elements from the inventory area Supply3 which are directly or indirectly dependent on T2Si1 are determined. This produces the subset T3Si2. To form T4Si3, inventory data elements from the inventory area Supply4 which are directly or indirectly dependent on data elements from T3Si2 and T2Si1 are combined in a further step Si3. Step Si4 forms a further subset T3Si4 in Supply3 which contains data elements which have at least one directly or indirectly dependent data element in T4Si3. T3Si4 is in turn completed such that directly and indirectly dependent elements from Supply3 can likewise be found in T3Si4. In a subsequent step Si5, a subset T2Si5 from Supply2 is formed. T2Si5 contains data elements which contain at least one directly or indirectly dependent data element in T3Si4 or T4Si3. In addition, T2Si5 is completed such that directly and indirectly dependent elements from Supply2 can likewise be found in T2Si5. For the set Supply1 containing all data elements for the reference data and metadata, no subset formation is undertaken. As indicated by the dashed arrow 15, steps Si2 to Si5 are repeated in an iterative process until it is no longer possible to associate any new, dependent data from the supply data inventories with the subsets. The final step Si6 contains creation of the supply file and/or of the supply data stream. To create the supply file or the supply data stream, the subsets T2Si1 and T2Si5 are combined, are sorted on the basis of their internal dependency and are stored as a file or data stream T2Si15-F. Similarly, the subsets T3i and T3e are combined, are sorted on the basis of their internal dependency and are stored as a file or data stream T3Si24-F. The subset T4i is likewise sorted on the basis of the internal dependency and stored as a file or data stream T4Si3-F. The set Supply1 is stored as a file or data stream Supply1-F. The supply file or the supply data stream IpsSupplyOut now contains the parts Supply1-F, T2Si15-F, T3Si24-F and T4Si3-F in this order.

In contrast to the example shown in figure 5, figure 6 shows how the completeness formation is performed using financial instrument identifications.

Figure 6 schematically shows the completeness formation for inventory data elements in the case of a prescribed list of primary keys Fps from the financial instruments area B3. In a first step Sf1, the data elements from the inventory area Supply3 of the financial instruments area B3 which contain one or more of the keys from the list Fps as primary key are selected. In the same step, these data elements have directly and indirectly dependent data elements from the inventory area Supply3 added to them, so that the subset T3Sf1 is produced. In a further step Sf2, the data elements from the inventory area Supply4 area B4 (events) which are directly or indirectly dependent on T3Sf1 are determined. This produces the subset T4Sf2. In step Sf3, a further subset T3Sf3 from the Supply3 inventory area is formed which contains data elements which have at least one directly or indirectly dependent data element in T4Sf2. T3Sf3 is in turn complete such that directly and indirectly dependent elements from the Supply3 inventory area can likewise be found in T3Sf3. In a subsequent step Sf4, a subset T2Sf4 of Supply2 is formed. T2Sf4 contains data elements which contain at least one directly or indirectly dependent data element in T3Sf3 or T4Sf2. In addition, T2Sf4 is completed such that directly and indirectly dependent elements from the Supply2 inventory area can likewise be found in

T2Sf4. As in the example shown in figure 5, it is also true here that the various steps are repeated iteratively until no new data elements can be associated with the subsets. For the Supply1 set containing all the data elements for the reference data and metadata, no subset formation is performed. The final step Sf5 contains creation of the supply file and/or the supply data stream. To create the supply file or the supply data stream, the subset T2Sf4 is sorted on the basis of the internal dependency and stored as a file or data stream T2Sf4-F. Similarly, the subsets T3 and T3Sf3 are combined, are sorted on the basis of the internal dependency and are stored as a file or data stream T3Sf13-F. The subset T4Sf2 is sorted on the basis of the internal dependency and stored as a file or data stream T4Sf2-F. The Supply1 set is stored on the basis of the internal dependency as a file or data stream Supply1-F. The supply file or the supply data stream IpsSupplyOut now contains the parts Supply1-F, T2Sf4-F, T3Sf13-F and T4Sf2-F in this order.

The two methods shown in figure 5 and figure 6 can be combined such that the completeness can be formed using a list of institution identifications (keys from institutions) and a list of financial instrument identifications (keys from financial instruments). For the sake of simplicity, the prices area B5 has not been taken into account in any of the examples which have been explained using figures 5 and 6. However, the methods explained can be expanded to an additional area 5 in similar fashion.

Figure 7 shows a schematic illustration of the combination of the completeness formations with primary keys from the areas B2 and B3. In line with figures 5 and 6, the supply files or supply data streams

IpsSupplyOut and FpsSupplyOut can be created with a set of primary keys from the areas B2 and B3. The two methods can be combined using the additional step S1c, so that a supply file or a supply data stream IfpsSupplyOut can be created for a set of primary keys from the areas B2 and B3. To this end, the subsets T2Si15-F, T3Si24-F and T4Si3-F are combined with T2Sf4-F, T3Sf13-F and T4Sf2-F, respectively, and are sorted on the basis of the internal dependency and stored as files T2-T2e-T2f-F, T3-T3i-T3e-T3f-F and T4i-T4Sf2-F. The s upply file or the supply data stream IfpsSupplyOut now contains the parts Supply1-F, T2-T2e-T2f-F, T3-T3i-T3e-T3f-F and T4i -T4Sf2-F in this order. IfpsSupplyOut is now the inventory data set which can be delivered to a customer, conditioned to his requirements using the method shown. In the same way, delta data elements are conditioned and delivered to customers. To keep the volume of data delivered as small as possible, only data which have undergone a change, are new or have been requested for the first time are delivered. Delivery of the inventory data set and of the associated delta data set with completeness formation, as shown in figures 5, 6 and 7, now allows automatic, computer-based processing of the data in situ without manual intervention by the customer, since the data are complete and their referential integrity is assured at all times. Besides the hierarchical structure of the data, delivery of the data sorted according to "deletion" and "mutation/new" also ensures referential integrity.

With customer needs for more complex data, such as data relating to financial instruments, which may arise from a right through capital increase, it may be that this method is not sufficient, however. For automatic processing, it is then necessary for basic data arising from rights through payment events likewise to be delivered. Figure 8 shows by way of example that by using data elements with payment events as a source for expanding the lists Fps containing financial instrument keys it becomes possible for portfolio information to be created automatically. Besides the list containing financial instrument keys, the customer can, in line with the invention, choose parameters which, in the inventive data processing device 10, cause data element types to be selected which serve as sources for the subsequently illustrated expansion. For the sake of simplicity, a file F1 with a single shares key f1 is shown by way of example. For the shares key f1, the completeness formation is performed in line with steps Sf1, Sf2 and Sf3, as shown in figure 6. Similarly for the sake of simplicity, only single data elements which are of interest in this example are shown. The data element TKFAD1 contains the basic data for the share with key f1. TKXCD1 is an event data element in the form of a rights distribution. That is to say that TKXCD1 contains a foreign key which is resolved using the basic data element of the right, TKFAD2. The data element TKXRD2 is the payment event for obtaining new shares using rights. This element is now not necessary in figures 4 to 6 for the processing which obtains referential integrity and therefore not in the data set which is formed on the basis of the key set F1. For many customers, however, the payment event for the purchase and the basic data TKFAD2 of the instrument which is to be purchased, in this case a new share, is significant. The inventive data processing device now allows an initial command, which can be executed by any customer, to be used to write foreign keys from financial instruments which appear in particular event data element types to a second, further key list S1 automatically during each completeness formation. After every pass of the methods in figures 5 to 7, the methods from figures 5 to 7 are applied to S1 or to another, newly created key list, and the resultant data set is delivered to the customer. In the example of the rights distribution, the second list S1 is created at step Sf2 and the foreign key for the right f2 is written to the list S2. The subsequent, second implementation of the method from figure 5 will include an exercise event TKXRD and an underlying TKXUD for the right. The underlying and the exercise event contain a foreign key for the financial instrument which is to be purchased. The basic element TKFAD3 is thus likewise included in the delivery set for the new share.